Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Discrete Mathematics 306 (2006) 1177 – 1188

DISCRETE MATHEMATICS

www.elsevier.com/locate/disc

Acyclically pushable bipartite permutation digraphs: An algorithm ☆

Romeo Rizzi

Dipartimento di Matematica ed Informatica (DIMI), Università di Udine, Via delle Scienze 208, I-33100 Udine, Italy

Received 2 May 2002; received in revised form 4 November 2005; accepted 29 November 2005
Available online 18 April 2006

Abstract

Given a digraph $D=(V,A)$ and an $X \subseteq V$, D^X denotes the digraph obtained from D by reversing those arcs with exactly one end in X. A digraph D is called *acyclically pushable* when there exists an $X \subseteq V$ such that D^X is acyclic. Huang, MacGillivray and Yeo have recently characterized, in terms of two excluded induced subgraphs on 7 and 8 nodes, those bipartite permutation digraphs which are acyclically pushable. We give an algorithmic proof of their result. Our proof delivers an $O(m^2)$ time algorithm to decide whether a bipartite permutation digraph is acyclically pushable and, if yes, to find a set X such that D^X is acyclic. (Huang, MacGillivray and Yeo's result clearly implies an $O(n^8)$ time algorithm to decide but the polynomiality of constructing X was still open.)

We define a *strongly acyclic digraph* as a digraph D such that D^X is acyclic for every X. We show how a result of Conforti et al [Balanced cycles and holes in bipartite graphs, Discrete Math. 199 (1–3) (1999) 27–33] can be essentially regarded as a characterization of strongly acyclic digraphs and also provides linear time algorithms to find a strongly acyclic orientation of an undirected graph, if one exists. Besides revealing this connection, we add simplicity to the structural and algorithmic results first given in Conforti et al [Balanced cycles and holes in bipartite graphs, Discrete Math. 199 (1–3) (1999) 27–33]. In particular, we avoid decomposing the graph into triconnected components.

We give an alternate proof of a theorem of Huang, MacGillivray and Wood characterizing acyclically pushable bipartite tournaments. Our proof leads to a linear time algorithm which, given a bipartite tournament as input, either returns a set X such that D^X is acyclic or a proof that D is not acyclically pushable.

Keywords: Acyclically pushable; Orientation; Push; Acyclic digraph

1. Introduction

The operation of *pushing vertices* of a digraph has received considerable attention in the past few years. Let $D=(V,A)$ be a simple digraph, and $X \subseteq V$. When X is *pushed*, the orientation of every arc with precisely one endpoint in X is reversed. The resulting digraph is denoted by D^X. An example of pushing is shown in Fig. 2. This operation can be used to define an equivalence relation: when D' can be obtained from D by pushing, then we write $D' \sim D$ and say that the two digraphs are (*push*) *equivalent*. (Transitivity follows since $(D^X)^Y = D^{X\Delta Y}$.) The equivalence class that

☆ Research partially done while enjoying hospitality at BRICS, Department of Computer Science, University of Aarhus, Denmark.
E-mail address: Romeo.Rizzi@dimi.uniud.it.

0012-365X/$ - see front matter
doi:10.1016/j.disc.2005.11.027

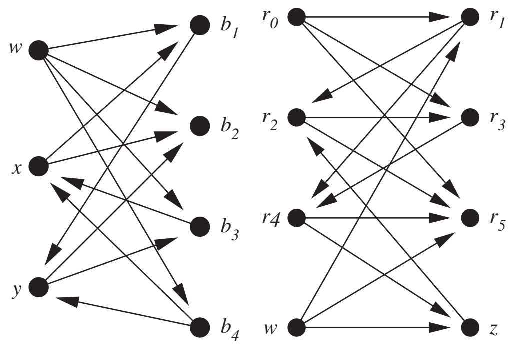

Fig. 1. M_1 and M_2, the two forbidden bipartite permutation digraphs.

contains D is denoted by $[D]$. A digraph which is equivalent to an acyclic digraph is called *acyclically pushable*. In [7], Klostermeyer proved that recognizing acyclically pushable digraphs is NP-complete and gave a superclass of the outerplanar digraphs whose members are all acyclically pushable. He also observed the following fact.

Fact 1 (*Klostermeyer [7], Huang et al. [5,6]*). *The property of being acyclically pushable is closed under taking subgraphs, under pushing, and under inverting the direction on all the arcs.*

A *tournament* is a simple digraph where any two nodes are adjacent. A *k-partite tournament* is a simple digraph where the nodes are partitioned into k disjoint classes and any two nodes are adjacent if and only if they belong to different classes. In [9], MacGillivray and Wood proved that a tournament D is not acyclically pushable if and only if the tournament $(\{a, b, c, v\}, \{ab, bc, ca, va, vb, vc\})$ can be obtained from D by pushing and node removals (*proof*: choose an arbitrary node v, push as to make v a source and observe that, if the tournament you have so obtained is not acyclic, then it contains a directed triangle).

In [5], Huang et al. gave a similar characterization for bipartite tournaments and for multipartite tournaments. (A *multipartite tournament* is a k-partite tournament for some $k \geqslant 2$.) Recently, in [6], Huang et al. extended the characterization in [9] to chordal digraphs (*hint to their proof*: every chordal graph has a simplicial node), proved that recognizing acyclically pushable bipartite digraphs is NP-complete, obtained Theorem 2 here below, and gave an infinite family of chordal bipartite digraphs whose members are all minimal not acyclically pushable (under taking subgraphs). A graph is *chordal* (resp., *chordal bipartite*) if it contains no induced cycle of length greater than three (resp., is bipartite and contains no induced cycle of length greater than four). A digraph is called *bipartite* (resp., *chordal*, *chordal bipartite*, *outerplanar*, *connected*) if its underlying graph is bipartite (resp., chordal, chordal bipartite, outerplanar, connected). A bipartite digraph $D = (U, V; A)$ is a *bipartite permutation digraph* if there exist node orderings $u_1, u_2, \ldots, u_{n_1}$ of U and $v_1, v_2, \ldots, v_{n_2}$ of V such that, for any $i < j$ and $h < k$, the following implication holds: if u_i and v_k are adjacent, and u_j and v_h are adjacent, then u_i and v_h are adjacent, and u_j and v_k are adjacent.

The main result in [6] is the following characterization of those bipartite permutation digraphs which are acyclically pushable, where M_1 and M_2 are the two digraphs depicted in Fig. 1.

Theorem 2. *A bipartite permutation digraph is acyclically pushable if and only if it does not contain any digraph in* $[M_1] \cup [M_2]$.

1.1. Our contribution

In Section 3, we give an alternate proof of a theorem of Huang, MacGillivray and Wood characterizing acyclically pushable bipartite tournaments. Our proof leads to a linear time algorithm which, given a bipartite tournament as input, either returns a set X such that D^X is acyclic or a proof that D is not acyclically pushable.

In [6], Huang et al. observed that Theorem 2 implies an $O(n^8)$ time algorithm to recognize acyclically pushable bipartite permutation digraphs and asked for a polynomial time algorithm which, given an acyclically pushable bipartite permutation digraph D, returns a set X such that D^X is acyclic. In Section 4, we give an algorithmic proof of Theorem 2. Our proof delivers an $O(m^2)$ time algorithm to decide whether a bipartite permutation digraphis acyclically pushable and, if yes, to find a set X such that D^X is acyclic.

In Section 5, we define a *strongly acyclic digraph* as a digraph D such that D^X is acyclic for every X. We show how a previous result of Conforti et al. [1] can be regarded, in disguise, as a characterization of strongly acyclic digraphs and also provides linear time algorithms to find a strongly acyclic orientation of a graph, if one exists. Besides revealing this connection, our treatment of this issue adds simplicity to the structural and algorithmic results first given in [1]. In particular, while the recognition algorithm given in [1] rests on Hopcroft and Tarjan's algorithm [4] to decompose a graph into triconnected components, our simpler algorithm does not need this non-trivial subroutine since our structural results directly apply also to graphs which are only biconnected. Furthermore, we propose elementary constructions for the classes of graphs considered in Section 5.

2. Preparatory lemmata and terminology

We are interested exclusively in *simple* digraphs, hence loops and anti-parallel arcs are banned from this paper. Let $D=(V,A)$ be a digraph and $u, v \in V$ two nodes of D. When $uv \in A$ is an arc of D, we say that u *dominates* v or v *is dominated by* u, and also say that u and v are *adjacent*. We shall use $I(v)$ (resp., $O(v)$) to denote the set of all vertices that dominate (resp., are dominated by) v. We shall use $N(v)$ to denote $I(v) \cup O(v)$ and $N[v]$ to denote $N(v) \cup \{v\}$. Moreover, the nodes in $I(v)$ (resp., $O(v)$, $N(v)$) are called the *in-neighbors* (resp., *out-neighbors*, *neighbors*) of v. The set $I(v)$ (resp., $O(v)$, $N(v)$) is called the *in-neighborhood* (resp., *out-neighborhood*, *neighborhood*) of v. A node v is called a *source* (resp., *sink*) when $I(v)$ (resp., $O(v)$) is empty. As customary, n denotes the number of nodes and m the number of arcs in the digraph of interest. We adopt the common usage notation $D=(U, V; A)$ to refer to bipartite digraphs, where U and V specify the color classes. In this case, $n_1 := |U|$, $n_2 := |V|$, and $n = n_1 + n_2$. When $S \subseteq V$, then $D\langle S\rangle$ denotes *the subgraph of D induced by S*, that is, the digraph obtained from D by removing the nodes in $\overline{S} = V \backslash S$ as well as all the arcs with at least one endpoint in $\overline{S}$. The same digraph could have also been indicated with $D\backslash\overline{S}$. As a special case, the digraph $D\backslash\{v\}$ obtained from D by the removal of a single node v deserves the shorthand notation $D\backslash v$. Suppose D is acyclic. Then the nodes of D can be linearly ordered as $v_1, v_2, \ldots, v_n$ such that v_i dominates v_j only if $i<j$. Such a linear order is called a *topological sort* of the acyclic digraph D. We will need the following lemmata.

Lemma 3 (*Huang et al. [5,6]*). *Let D be an acyclically pushable digraph and let u be a vertex of D. Then there exists $X \subseteq V\backslash\{u\}$ such that D^X is acyclic and u is a source in D^X.*

Proof. The condition $u \notin X$ is easy to meet since pushing X is the same as pushing $V\backslash X$. Note also that when $D' \in [D]$ is acyclic and $v_1, v_2, \ldots, v_i = u, \ldots, v_n$ is a topological sort of D', then pushing $\{v_1, v_2, \ldots, v_{i-1}\}$ will not destroy acyclicity while making u a source. □

Lemma 4. *Let $D=(V,A)$ be a minimal not acyclically pushable digraph. Then every node of D has degree at least* 3.

Proof. Let v be a node of degree at most 2 in D. Let $X \subseteq V\backslash\{v\}$ such that $(D\backslash v)^X$ is acyclic. Assume D^X contains a directed cycle C. Necessarily, C must go through v, hence v has degree 2. Let (s,v) and (v,t) be the two arcs of D^X incident with v. So, C contains a t,s-path contained in $(D\backslash v)^X$. Since $(D\backslash v)^X$ is acyclic, then $(D\backslash v)^X$ contains no s,t-path. Hence, $D^{X\cup\{v\}}$ is acyclic. □

Let u and v be two non-adjacent nodes. We say that node u *subsumes* node v if either $I(v) \subseteq I(u)$ and $O(v) \subseteq O(u)$ both occur, or $I(v) \subseteq O(u)$ and $O(v) \subseteq I(u)$ both occur. Note that if u subsumes v in D then u subsumes v in every digraph in $[D]$.

Lemma 5. *Let $D=(V,A)$ be a minimal not acyclically pushable digraph. Then no node of D subsumes another node of D.*

Proof. Assume that u subsumes v and that there exists an $X \subseteq V\backslash\{v\}$ for which $(D\backslash v)^X$ is acyclic. Let X' be the one set among X and $X \cup \{v\}$ such that $I(v) \subseteq I(u)$ and $\mathrm{O}(v) \subseteq \mathrm{O}(u)$ both occur in $D^{X'}$. We claim that $D^{X'}$ is acyclic. Indeed, $D^{X'}\backslash v = (D\backslash v)^X$ is acyclic, and a topological sort of $D^{X'}\backslash v$ can be extended to a topological sort of $D^{X'}$ by inserting v just before, or just after, u. Therefore, $D^{X'}$ is also acyclic. $\square$

3. Acyclically pushable bipartite tournaments

Let M_1 be the bipartite tournament depicted in Fig. 1 on the left. It can be checked by simple inspection of cases that M_1 is not acyclically pushable. (By Lemma 3, applied with $u = w$, only the four cases with $X \subseteq \{x, y\}$ need to be checked.) In this section, we give an alternate proof of the following theorem of Huang, MacGillivray and Wood.

Theorem 6 (*Huang et al.* [5]). *Let D be a bipartite tournament. Then D is acyclically pushable if and only if it does not contain any digraph in* $[M_1]$.

Our proof implies a linear time algorithm to decide whether a given input bipartite tournament D is acyclically pushable. In case the answer is YES, then the algorithm also returns a set X such that D^X is acyclic. In case the answer is NO, then the algorithm returns an induced subgraph of D which is isomorphic to a digraph in $[M_1]$. In both cases, the running time is bounded from above by a linear function in the number of arcs of D, that is, by a linear function in the length of the input.

Step 1: Let $D = (U, V; A)$ be a bipartite tournament. Let $n_1 = |U|$ and $n_2 = |V|$. Choose any node w in U. By pushing $I(w)$, we can assume that w is a source of D. By Lemma 5, we can also assume that w is the only source or sink in U. All this can be done in linear time.

At this point, before introducing the next step, let us spend a first consideration at the base of the approach.

Two nodes $x, y \in U$ are called *comparable* if $\mathrm{O}(x) \subseteq \mathrm{O}(y)$, or $\mathrm{O}(x) \subseteq I(y)$, or $I(x) \subseteq \mathrm{O}(y)$, or $I(x) \subseteq I(y)$. The following fact was proved in [5].

Lemma 7 (*Huang et al.* [5]). *Let* $w \in U$ *have* $I(w) = \emptyset$. *Suppose that* $U\backslash\{w\}$ *contains two not comparable nodes x and y. Then the bipartite tournament D contains an* M_1 *as an induced subgraph.*

Proof. Let $b_1 \in \mathrm{O}(x) \cap I(y)$, $b_2 \in \mathrm{O}(x) \cap \mathrm{O}(y)$, $b_3 \in I(x) \cap \mathrm{O}(y)$ and $b_4 \in I(x) \cap I(y)$. Then the subgraph of D induced by $\{w, x, y, b_1, b_2, b_3, b_4\}$ is precisely the digraph M_1 depicted in Fig. 1 on the left. $\square$

Step 2: For every node u in $U\backslash\{w\}$, compute the values $in_u := |I(u)|$, $out_u := |\mathrm{O}(u)|$, and $\phi_u := \max\{in_u; out_u\}$. Let x be a node in $U\backslash\{w\}$ such that ϕ_x is maximum.

Step 3: Push x if $out_x < in_x$ (and update in_x and out_x accordingly). After this, we have that $out_x \geqslant in_x$. Moreover, we can assume that for no node $u \in U\backslash\{w, x\}$, $\mathrm{O}(x) \subseteq \mathrm{O}(u)$ or $\mathrm{O}(x) \subseteq I(u)$ occurs, after we exclude the cases of strict equality by resorting to Lemma 5.

Step 4: For every node u in $U\backslash\{w, x\}$, check whether u and x are comparable. If some such u is found not comparable with x, then answer no and return an induced subgraph of D isomorphic to M_1 as indicated in the (constructive) proof of Lemma 7.

Assume therefore that u and x are comparable for all u in $U\backslash\{w, x\}$. By what established in step 3, only two cases can occur: either $I(x) \subset \mathrm{O}(u)$, or $I(x) \subset I(u)$. In the first case, push u (and update in_u and out_u accordingly). At this point, and in both cases, we have that $\mathrm{O}(u) \subset \mathrm{O}(x)$. Doing this for every node u in $U\backslash\{w, x\}$ takes linear time in total.

Lemma 8. *Let* $w \in U$ *have* $I(w) = \emptyset$. *Let* $x \in U$ *have* $\mathrm{O}(x) \subset \mathrm{O}(w)$. *Consider two nodes* α *and* β *in* $U\backslash\{w, x\}$ *and assume* $\mathrm{O}(\alpha), \mathrm{O}(\beta) \subset \mathrm{O}(x)$. *Assume further that* $\mathrm{O}(\alpha) \cap \mathrm{O}(\beta) = \emptyset$. *Then D contains an induced subgraph in* $[M_1]$.

Proof. Let c be a node in $\mathrm{O}(w)\backslash\mathrm{O}(x)$. Let a be a node in $\mathrm{O}(\alpha)$ and b be a node in $\mathrm{O}(\beta)$. Then the subgraph of D induced by $\{w, \alpha, \beta, x, a, b, c\}$ is precisely the digraph M_1' depicted in Fig. 2 on the left. Moreover, as shown in the same figure, M_1' becomes M_1'' after pushing $\{w, x, \alpha\}$. Notice from the picture that we have also relabeled w as ω and x

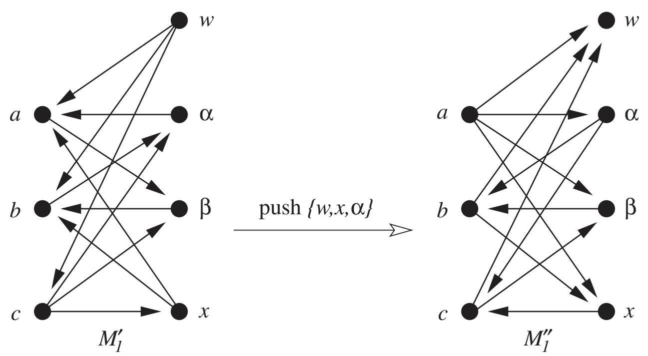

Fig. 2. From M'_1 to M''_1 by pushing $\{w, x, \alpha\}$.

as χ in preparation to the conclusive step: the claim now follows since the map $a \to w, b \to x, c \to y, \omega \to b_2, \alpha \to b_4, \beta \to b_3, \chi \to b_1$ is an isomorphism from M''_1 to M_1. □

Step 5: Sort the nodes in $U \backslash \{w, x\}$ as $u_1, u_2, \ldots, u_{n_1-2}$ so that $i < j$ implies $out_{u_i} \leqslant out_{u_j}$. This can actually be done in $O(n_1 + n_2)$ time by *Bucket Sort* (see [2]).

Step 6: For $i := 1$ to $n_1 - 3$, check whether $O(u_i) \subseteq O(u_{i+1})$. Assume that for some $\bar{\imath}$ we have that $O(u_{\bar{\imath}}) \nsubseteq O(u_{\bar{\imath}+1})$, then an induced subgraph of *D* which is isomorphic to a digraph in $[M_1]$ is returned as follows: if $O(u_{\bar{\imath}})$ and $O(u_{\bar{\imath}+1})$ are disjoint, then apply Lemma 8, otherwise, notice that $O(u_{\bar{\imath}+1}) \nsubseteq O(u_{\bar{\imath}})$ also holds since $out_{u_{\bar{\imath}+1}} \geqslant out_{u_{\bar{\imath}}}$ by the sorting imposed in *Step 5*. In this second case, $u_{\bar{\imath}}$ and $u_{\bar{\imath}+1}$ are not comparable, hence Lemma 7 applies.

Note that the whole for-cycle of the present step, in total, can be done in linear time. Indeed, there are $n_1 - 3$ set-inclusions to be checked and each check can be performed within $O(n_2)$ operations. Consider that the size of the input is $O(n_1 n_2)$ since *D* is a bipartite tournament and the direction of each arc of *D* needs to be specified by the input.

Step 7: Assume therefore that $O(u_i) \subseteq O(u_{i+1})$ holds for $i = 1, \ldots, n_1 - 3$. Then $O(u_1) \subseteq O(u_2) \subseteq \cdots \subseteq O(u_{n_1-2}) \subset O(x) \subset O(w)$. We claim that in this case *D* is acyclic. This assertion is certified by the existence of at least one topological sort of *D*. Indeed, where $\{S\}$ denotes any one of the $|S|!$ sorts of the nodes of an $S \subseteq V$, the (non-empty) family of topological sorts of *D* finds a compact description in the writing

$$w, \{O(w) \backslash O(x)\},\ x, \{O(x) \backslash O(u_{n_1-2})\},\ u_{n_1-2}, \ldots,\ u_2, \{O(u_2) \backslash O(u_1)\},\ u_1, O(u_1).$$

4. Algorithmic proof of Theorem 2

In this section, we give an algorithmic proof of Theorem 2. We consider this proof algorithmic in the sense that it leads to a polynomial time algorithm to decide whether a bipartite permutation digraph is acyclically pushable and, if yes, to find a set *X* such that D^X is acyclic. (We must here recall that Huang, MacGillivray and Yeo's result clearly implies an $O(n^8)$ time algorithm to decide but the polynomiality of constructing *X* was still open.)

Before entering into the proof, let us introduce some further notation specific to this section. When we know that two nodes, say *a* and *b*, are adjacent, but the orientation of the arc between *a* and *b* is not known to us, then we will speak of the $\{a, b\}$ *arc*. Within the proof, we will be arguing about certain 4-cycles at some point: a 4-*cycle* is a simple and closed directed walk traversing four arcs and four nodes. Notice that every non-acyclic bipartite tournament contains a 4-cycle. When we say that a, b, c, d is a 4-cycle of *D*, this means that ab, bc, cd, da are all arcs of *D*. When we say that $\{a, b, c, d\}$ is a 4-cycle of *D*, this means that either a, b, c, d or d, c, b, a is a 4-cycle of *D*.

Let $D = (U, V; A)$ be a connected bipartite permutation digraph. Let $u_0, u_1, \ldots, u_{n_1}$ and $v_0, v_1, \ldots, v_{n_2}$ be orderings of the nodes of *U* and *V*, respectively, satisfying the property in the definition of bipartite permutation digraph. That property ensures not only that u_0 and v_0 are adjacent, but also that for any node *v* in $N(u_0)$ and any node *u* in $N(v_0)$, the nodes *u* and *v* are adjacent. Moreover, by Lemma 4, we can assume that u_0 and v_0 at least have other two neighbors each. Let $S = D\langle N[u_0] \cup N[v_0] \rangle$. Then *S* is a bipartite tournament on color classes $U_S := N(v_0)$ and $V_S := N(u_0)$. By Theorem 6, we can push as to make *S* is acyclic, else *S* contains a digraph in $[M_1]$, which also means that *D* contains a digraph in $[M_1]$. We hence assume S to be acyclic in the following. By Lemma 3, we can also assume that u_0 is a source. Let $r_0, r_1, \dots, r_s$ be a topological sort of the nodes in S such that $r_0 = u_0$. (Such a topological sort exists since S is acyclic and u_0 is a source.) Let k be the integer such that $v_0 = r_k$. In principle, we could have assumed v_0 to be a source instead of u_0. To exploit also this degree of freedom, we consider two conditions which can possibly occur at this point:

(i) there exists a node $w \in U \backslash U_S$ and two integers q_1, q_2 with $q_1 < k < q_2$ such that w either dominates both r_{q_1} and r_{q_2} or is dominated by them both;
(ii) there exists a node $z \in V \backslash V_S$ and two integers p_1, p_2 with $p_1 < k < p_2$ such that z dominates r_{p_1} and is dominated by r_{p_2} or vice versa.

We claim that we can always assume w.l.o.g. that (ii) $\Rightarrow$ (i) is true in D. Indeed, assume to the contrary that (ii) holds while (i) does not. Push $\{r_0, r_1, \dots, r_{k-1}\}$. Clearly, $r'_i := r_{(i+k) \bmod (s+1)}$ is a topological sort of the nodes in S with respect to the new directions of the arcs. Now v_0 is a source and has taken the place of u_0. Indeed, by inverting the roles of U and V, note that (ii) $\Rightarrow$ (i) is now true since (i) holds (and (ii) does not).

Clearly, r_s is a sink in S. By Lemma 5, r_s belongs to V, otherwise r_s subsumes u_0 in D, since u_0 has no neighbor outside S. Analogously, r_1 belongs to V, else r_1 would be a source in S and subsume u_0.

Case 1: $v_0 = r_1$.

Let $\overline{m}$ be the greatest integer such that $u_{\overline{m}} \in U_S$. Let q be the integer such that $r_q = u_{\overline{m}}$. Note that $q > 1$. (Since $\overline{m} > 0$, then $u_{\overline{m}} \neq u_0 = r_0$, hence $q > 0$. Since $u_{\overline{m}} \in U$ whereas $r_1 = v_0 \in V$, then $q > 1$.) Let $\overline{D}$ be the digraph obtained from D by removing the arc $r_1 r_q = v_0 u_{\overline{m}}$. Note that $\overline{D}$ is a bipartite permutation digraph, as certified by the orderings $u_0, u_1, \dots, u_{n_1}$ of U and $v_0, v_1, \dots, v_{n_2}$ of V.

Recursion: We can find a node set X such that $\overline{D}^X$ is acyclic, unless we retrieve a digraph in $[M_1] \cup [M_2]$ contained in $\overline{D}$, and hence in D. By Lemma 3, we can assume $r_s \notin X$ is a sink in $\overline{D}^X$, hence $X \cap U_S = \emptyset$. Assume D^X contains a directed cycle, and hence a 4-cycle Q. Necessarily, $\{v_0, r_q\} \in Q$, even if we do not actually know whether $\{v_0, r_q\}$ will be directed from v_0 (if $v_0 \notin X$) or toward v_0 (if $v_0 \in X$). However, all arcs incident with v_0, except $\{u_0, v_0\}$, will be either all directed from v_0 (if $v_0 \notin X$) or all directed toward v_0 (if $v_0 \in X$). Therefore, $\{u_0, v_0\} \in Q$. The four nodes of Q are hence $u_0, v_0, r_q = u_{\overline{m}}$ and a node in V_S. Let p be the integer such that r_p is this fourth node. So, $Q = \{u_0, v_0, r_q, r_p\}$. Necessarily, $p < q$ otherwise $\{r_p, r_q\}$ and $\{r_p, u_0\}$ would either both exit or both enter r_p in D^X. Since S was acyclic and $u_0, r_q \notin X$, then $[(r_p \in X) \Leftrightarrow (v_0 \notin X)]$. Moreover, for no node r_h in $U_S \backslash \{r_q\}$ we can have $h > p$, since otherwise replacing r_q with r_h we would obtain a 4-cycle $\tilde{Q} = \{u_0, v_0, r_h, r_p\}$ already in $\overline{D}^X$. Let us pause for a moment just to state what we are striving to do: we want to add back $\{v_0, r_q\}$ to the acyclic $\overline{D}^X$. Maybe this introduces a directed cycle Q, but we have a weapon to help us: pushing v_0. Let Q' be a 4-cycle in $D^{X \varDelta \{v_0\}}$. Clearly, Q' must necessarily go through v_0 and hence contain $\{u_0, v_0\}$ as we argued for Q. Let $u_0, v_0, r_{\tilde{q}}, r_{\tilde{p}}$ the nodes as encountered going along Q' (either forward or backward). So, $Q' = \{u_0, v_0, r_{\tilde{q}}, r_{\tilde{p}}\}$. Clearly, $\tilde{p} < \tilde{q}$ otherwise $u_0 r_{\tilde{p}}$ and $r_{\tilde{q}} r_{\tilde{p}}$ would either both exit or both enter $r_{\tilde{p}}$. Necessarily, $\tilde{q} \neq q$, since otherwise $\overline{D}^X \backslash v_0 = D^X \backslash v_0$ would contain a 4-cycle made up by the two paths $Q \backslash v_0$ and $Q' \backslash v_0$. Actually, $\tilde{q} < p$, since we said that for no node r_h in $U_S \backslash \{r_q\}$ we can have $h > p$. As a consequence, $\tilde{p} < \tilde{q} < p$. Again, since S was acyclic and $u_0, r_{\tilde{q}} \notin X \varDelta \{v_0\}$, then $[(r_{\tilde{p}} \in X \varDelta \{v_0\}) \Leftrightarrow (v_0 \notin X \varDelta \{v_0\})]$, that is, $[(r_{\tilde{p}} \in X) \Leftrightarrow (v_0 \in X)]$. In conclusion, since $[(r_p \in X) \Leftrightarrow (v_0 \notin X)]$ had been shown above, then $[(r_p \in X) \Leftrightarrow (r_{\tilde{p}} \notin X)]$. We have finally obtained a contradiction: $r_{\tilde{p}}$ can take the place of v_0 in Q, that is, $\{u_0, r_{\tilde{p}}, r_q, r_p\}$ is also a 4-cycle of D^X, hence a 4-cycle of the acyclic $\overline{D}^X$, which is an absurd. This shows that either X or $X \varDelta \{v_0\}$ works fine.

Case 2: $v_0 = r_s$.

We get back to Case 1 by pushing $U_S \backslash \{u_0\}$. Note that $S^{U_S \backslash \{u_0\}} = (S^{u_0})^{U_S}$ is acyclic since u_0 is a source in S, whereas U_S is a side of the bipartition of S^{u_0}.

Case 3: $v_0 \neq r_1, r_s$.

Recursion: Let $\underset{,}{D} := D \backslash v_0$. We can find a node set X such that $\underset{,}{D}^X$ is acyclic, unless we retrieve a digraph in $[M_1] \cup [M_2]$ contained in $\underset{,}{D}$, and hence in D. By Lemma 3, we can assume $u_0 \notin X$ is a source in $\underset{,}{D}^X$, hence $X \cap N[u_0] = \emptyset$. Since every node in $U_S \backslash \{u_0\}$ is dominated by r_1 and dominates r_s, either $X \cap U_S = \emptyset$ or $U_S \backslash \{u_0\} \subseteq X$, as otherwise $\underset{,}{D}^X$ contains a 4-cycle (indeed, r_1, a, r_s, b are the nodes of a 4-cycle for any $a \in (U_S \backslash \{u_0\}) \backslash X$ and $b \in U_S \cap X$). At this point, we can assume w.l.o.g. that X contains no node of S. Indeed, in case $U_S \backslash \{u_0\} \subseteq X$, we can always consider the symmetric difference $X' = X \Delta (U \backslash \{u_0\})$. (Pushing u_0 cannot destroy acyclicity since u_0 is a source and pushing U cannot destroy acyclicity since U is a side of the bipartition.)

To conclude the proof, we will now show that either D^X is acyclic or D contains a subgraph in $[M_2]$. Indeed, assume D^X is not acyclic and let Q be any 4-cycle in D^X. Since both S and $\c{D}^X$ are acyclic, and no node of S is in X, then Q traverses both $v_0 = r_k$ and a node not in S. Let r_k, r_{p_2}, z, r_{p_1} be the nodes as encountered going along Q. So, $Q = r_k, r_{p_2}, z, r_{p_1}$. Hence, $p_1 < k < p_2$ and z dominates r_{p_1} and is dominated by r_{p_2} in D^X. Thus property (ii) holds w.r.t. D^X. Since no node of S is in X, properties (i) and (ii) are not affected by pushing X, that is, they hold w.r.t. D^X if and only if they hold w.r.t. D. Hence, (ii) holds both w.r.t. D^X and w.r.t. D, and since (ii) $\Rightarrow$ (i) is true in D, then (i) holds both w.r.t. D and w.r.t. D^X. We conclude that in D^X there exists a node $w \in U \backslash U_S$ and integers q_1, q_2 with $q_1 < k < q_2$ such that w either dominates both r_{q_1} and r_{q_2} or is dominated by them both. Consider the subgraph R of D^X induced by $\{u_0 = r_0, v_0 = r_k, z, w, r_{p_1}, r_{p_2}, r_{q_1}, r_{q_2}\}$. By possibly pushing w, we can assume that w dominates bothr_{q_1} and r_{q_2}. Algorithmically, let $R' := R$ if w dominates both r_{q_1} and r_{q_2} and let $R' := R^{\{w\}}$ otherwise. We claim that R' is isomorphic to M_2. Indeed, $q_1 < p_1$ (and $p_2 < q_2$), since otherwise $\c{D}^X$ would contain the 4-cycle $\tilde{Q} = r_{q_1}, r_{p_2}, z, r_{p_1}$ (or $r_{q_2}, r_{p_2}, z, r_{p_1}$) obtained from Q by substituting r_k by r_{q_1} (or r_{q_2}). Hence, $0 < q_1 < p_1 < k < p_2 < q_2$, which already forces the right direction for all the arcs in S. It remains to force the right direction for $\{w, z\}$, which must be present in D since D is a bipartite permutation digraph. And indeed, the wrong direction zw is forbidden, since otherwise z, w, r_{q_1}, r_{p_2} (or r_{p_1}, r_{q_2}, w, z, if w has been pushed when going from R to R') would have been a 4-cycle in $\c{D}^X$.

Remark 9. We consider the above proof algorithmic in the sense that it leads to a polynomial time algorithm to decide whether a bipartite permutation digraph is acyclically pushable and, if yes, to find a set X such that D^X is acyclic. The algorithm resorts to at most m calls to a subroutine solving the same problem for bipartite tournaments, as for example the one from Section 3. One can assess, after some checking, that an $O(m^2)$ implementation of the algorithm is possible.

5. Strongly acyclic digraphs

A *strongly acyclic digraph* is a digraph D such that D^X is acyclic for every X. In this section, we show how strongly acyclic digraphs can be recognized and how, given any undirected graph G, we can decide whether G admits a strongly acyclic orientation and, if so, produce one.

In this section, the *underlying graph* G_D of a digraph D is the undirected graph obtained from D by simply disregarding the orientation of the arcs, that is, G_D has the same node set as D and two nodes are adjacent in G_D precisely when they are adjacent in D. A *round* of D is any subdigraph D' of D such that $G_{D'}$ is an undirected cycle. A *cut-node* of a graph is a node whose removal disconnects the graph. Those readers who want to know more about terms like *biconnected*, *triconnected*, *blocks*, *subdivision*, and *series-parallel*, should refer to [10].

The following observation shows that whether D is strongly acyclic or not only depends on the single rounds it contains.

Fact 10. *A digraph is strongly acyclic if and only if all its rounds are strongly acyclic.*

When G_D contains a cycle of odd length, then there exists always an X such that D^X contains a directed cycle. Indeed, this was first observed in [7] and is also explained in Fig. 3.

The fact that odd cycles have no strongly acyclic orientations is also a corollary of the following lemma, which was implicit in [8] in the discussion preceding Lemma 2 (in [8]).

Lemma 11. *Let $\overline{C}_1$ and $\overline{C}_2$ be two orientations of a same undirected cycle. Then $[\overline{C}_1] = [\overline{C}_2]$ if and only if $\overline{C}_1$ and $\overline{C}_2$ differ on the orientation of an even number of arcs.*

Proof. Pushing any node in an orientation of a cycle reverses precisely two arcs. This implies that if $\overline{C}_2 \in [\overline{C}_1]$, then $\overline{C}_1$ and $\overline{C}_2$ differ on the orientation of an even number of arcs.

Assume now that $\overline{C}_1$ and $\overline{C}_2$ differ on the orientation of an even number of arcs. Let $\overline{P}$ be the directed path obtained from $\overline{C}_1$ by removing one single arc. As argued in Fig. 3, we can push a suitably chosen set of nodes in $\overline{C}_2$ to obtain a

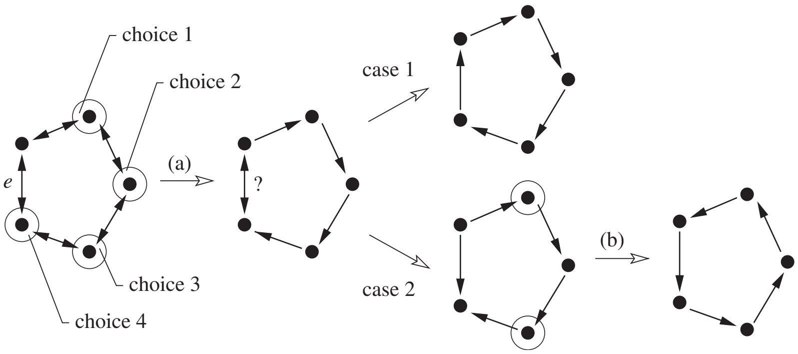

Fig. 3. First, in (a), push some of the circled nodes as to make $\overline{C}\backslash e$ a directed path. Next, in case e is directed the wrong way, then push as in (b) the smallest color class of $\overline{C}\backslash e$. How to push as to make any odd cycle $\overline{C}$ directed.

push equivalent orientation $\widetilde{C}_2$ containing $\overline{P}$. Notice that $\widetilde{C}_2$, $\overline{C}_2$ and $\overline{C}_1$ all differ on the orientation of an even number of arcs. Since $\overline{C}_1$ and $\widetilde{C}_2$ differ on the orientation of at most one arc, then $\widetilde{C}_2 = \overline{C}_1$. □

Corollary 12 (*Klostermeyer [7], Klostermeyer and Soltés [8]*). *No orientation of an odd cycle C is strongly acyclic.*

Proof. Let $\overrightarrow{C}$ and $\overleftarrow{C}$ be the only two directed cycles which are orientations of C. Then $\overrightarrow{C}$ and $\overleftarrow{C}$ differ on the orientation of $|C|$ arcs, which is an odd number of arcs. It follows that any orientation of C belongs to $[\overrightarrow{C}] \cup [\overleftarrow{C}]$. □

Therefore, only bipartite digraphs can be strongly acyclic and only bipartite graphs can have strongly acyclic orientations. Hence, we need to settle only the bipartite case. As shown in Section 5.1, the analysis of the bipartite case was essentially done, though in a disguised form, in a paper by Conforti et al. [1]. Section 5.1 aims at providing the means for translating back and forth from the formulation proposed in [1] and the one here considered. But now, back to our formulation, we will show how to face the same problems directly into our setting. Our self-contained exposition will partly borrow-translate from [1] and partly follow a shorter route based on a sharper structural decomposition viewpoint (see Theorem 18). These improvements lead to simpler algorithms than those proposed in [1].

Given an undirected cycle C, let $\overrightarrow{C}$ and $\overleftarrow{C}$ be the only two directed cycles which are orientations of C. Assume C is an even cycle. Then, $[\overrightarrow{C}] = [\overleftarrow{C}]$ follows from Lemma 11 but should also be clear if one considers to push one of the two color classes of C. Moreover, for any orientation $\overline{C}$ of C, the parity of the number of arcs whose orientation in $\overline{C}$ agrees with the orientation of $\overrightarrow{C}$ and the parity of the number of arcs whose orientation in $\overline{C}$ agrees with the orientation of $\overleftarrow{C}$ are the same. We say that an orientation $\overline{C}$ of C is *odd* (resp., *even*) if this parity is odd (resp., even). The following corollary of Lemma 11 offers an understanding of orientations of even cycles. This corollary corresponds to Theorem 8 in [7].

Corollary 13 (*Klostermeyer [7]*). *An orientation $\overline{C}$ of an even cycle is strongly acyclic if and only if $\overline{C}$ is odd, that is, if and only if an odd number of arcs are directed in each direction.*

It was observed in [1] that the problem of deciding whether an input digraph D is strongly acyclic can be reduced to the problem of finding a strongly acyclic orientation of G_D. (Clearly, all arguments in [1] are framed in a different setting and we assume a proper understanding of such assertions whose rigour only follows from the direct connection shown in Section 5.1.) Indeed, if G_D admits no strongly acyclic orientation, then surely D is not strongly acyclic. Assume therefore that G_D admits a strongly acyclic orientation D'. Let T be any spanning tree of G_D. (We can w.l.o.g. assume that G_D is connected.) Clearly, any digraph obtained from D' by pushing is strongly acyclic. Therefore, we can always assume that the orientations of the arcs in D and in D' agree on T, since for every edge e in T there is a set of nodes X such that e is the only edge of T with an odd number of endnodes in X. (Take as X any one of the two connected components of $T\backslash e$.) At this point, D is strongly acyclic if and only if $D = D'$, since for every edge e of G_D not in T there exists a cycle in $T \cup \{e\}$ and, by Lemma 11, two orientations of a cycle cannot be both strongly acyclic if they differ on the orientation of a single edge.

To introduce the characterization of the bipartite graphs G that admit strongly acyclic orientations given in [1], consider two nodes u, v of G. A 3-*path configuration* connecting u and v, denoted by $3PC(u, v)$, is defined by three internally node-disjoint paths, each connecting u and v. A $3PC(u, v)$ is *homogeneous* if u and v belong to the same side of the bipartition. By Corollary 13, a homogeneous $3PC(u, v)$ admits no strongly acyclic orientation, since at least one among its three cycles must be given an even orientation. Since being strongly acyclic is closed under taking subgraphs, any graph containing a homogeneous $3PC(u, v)$ admits no strongly acyclic orientation. In this section, we give a simple linear time algorithm, which, given an undirected bipartite graph G as input, outputs either a strongly acyclic orientation of G or a homogeneous 3-path configuration contained in G. This will prove the following theorem.

Theorem 14 (*Theorem 3.3 in [1]*). *A bipartite graph G admits a strongly acyclic orientation if and only if G does not contain a homogeneous 3-path configuration.*

Given a cut-node x of a graph G, let $C_1, C_2, \ldots, C_k$ be the connected components of $G \backslash x$. The *blocks of the cut-node decomposition of G by x* are the graphs $G_1, G_2, \ldots, G_k$, where each G_i is the subgraph of G induced by the node set $V(C_i) \cup \{x\}$. The reason why we can assume that G is biconnected is expressed by the following simple lemma.

Lemma 15 (*Lemma 4.4 in [1]*). *Let $G_1, G_2, \ldots, G_k$ be the blocks of a cut-node decomposition of a graph G. Then G admits a strongly acyclic orientation if and only if each block G_i does.*

To achieve a linear time algorithm, we employ a linear time routine to decompose G into blocks. See [2] for an in-depth exposition of a classical DFS solution to this problem. Unlike in [1], we avoid resorting to the more involved and sophisticated Hopcroft and Tarjan's algorithm [4] to decompose G into triconnected components.

A biconnected graph G is called *series-parallel* if G can be obtained from a cycle C by repeated applications of the following two operations:

1. replace an edge with two parallel edges,
2. subdivide an edge into a path by introducing new nodes.

Moreover, given a series-parallel graph G, a construction of G as above is called a *series-parallel construction of G* and can be easily obtained in linear time. (Indeed, every series-parallel graph has either two parallel edges or a node of degree 2.)

Notice that any bipartite subdivision of K_4 (i.e. any bipartite graph which can be obtained starting from the complete graph on four nodes by applying only operations of type 2) contains a homogeneous 3-path configuration since two out of the four degree-three nodes of the subdivision must necessarily be on a same side of the bipartition. Therefore, as done in [1] (see Theorem 4.2), we can restrict our attention to series-parallel graphs, by the following classical result of Dirac:

Theorem 16 (*Dirac [3]*). *A biconnected graph is series-parallel if and only if it has no K_4 minor.*

Our original approach consists in introducing a special and restricted form of series-parallel construction for biconnected bipartite graphs, paying attention to allow only those operations which maintain the property of having a strongly acyclic orientation, and discovering in the end that the proposed construction yields precisely those biconnected bipartite graphs containing no homogeneous 3-path configuration.

A subdivision (operation of type 2) is called *even* when the number of nodes it introduces is even, that is, when a copy of uv is replaced by a path of odd length. The importance of even subdivisions is explained by the following lemma.

Lemma 17. *Let uv be an arc in a digraph D and let D' be a digraph obtained from D by subdividing arc uv as to obtain a directed path from u to v of odd length. If D is strongly acyclic, then D' is strongly acyclic.*

Proof. This follows from Corollary 13, and by Fact 10. □

A series-parallel construction of G is called *even* when it starts from an even cycle C and involves only even subdivisions. Clearly, when G admits an even series-parallel construction, then G admits a strongly acyclic orientation. Indeed, a strongly acyclic orientation of G is obtained when we start from any strongly acyclic orientation of C (which exists by Corollary 13, since C is even) and mimic the construction of G as follows:

1′. when in the construction of G an edge is replaced with two parallel edges, then replace the corresponding arc, say uv, with two parallel arcs with tail u and head v,
2′. when in the construction of G an edge is subdivided, then subdivide the corresponding arc, say uv, by introducing the same set of internal nodes so as to obtain a directed path from u to v.

Therefore, only biconnected bipartite graphs containing no homogeneous 3-path configuration can have an even series-parallel construction. We will now obtain the following original structural result. Theorem 14 will follow as a corollary.

Theorem 18. *Every biconnected bipartite graph containing no homogeneous* 3*-path configuration admits an even series-parallel construction.*

To prove Theorem 18 without rediscovering Dirac's theorem, we describe a process which, given a series-parallel construction of a biconnected bipartite series-parallel graph G, either converts the given construction into an even one or finds out an homogeneous 3-path configuration in G.

To begin with, notice that even if $G=(U,V;E)$ is bipartite, the intermediate graphs in the given series-parallel construction of G may not be bipartite. Nevertheless, the nodes of the intermediate graphs can all be regarded as nodes of G, hence we can assign them a color class and say whether they belong to U or V. A series-parallel construction of G is called *bipartite* if whenever an edge ab is replaced by two parallel edges, then a and b are of opposite colors in G. This new notion is motivated by the following lemma.

Lemma 19. *If a series-parallel construction for a biconnected bipartite graph G is not bipartite, then G contains a homogeneous* 3*-path configuration.*

Proof. Note that every intermediate graph appearing along the construction of G is biconnected. Hence, before edge ab was replaced by two parallel edges, there already were two internally node-disjoint paths between a and b. After the replacement, the number of internally node-disjoint paths between a and b has gone up to three. And afterward, it can never decrease. Since a and b are in the same color class, G contains a homogeneous $3PC(a,b)$. □

Unfortunately, not every bipartite series-parallel construction is even. We can however enforce this desirable implication by restricting our attention to "canonical" series-parallel construction in which no nested subdivisions occur. Formally, a *canonical* series-parallel construction starts from a cycle C of *unmarked* edges and iteratively applies one of the following three operations.

1. replace an edge with two parallel *marked* edges,
2. subdivide a *marked* edge into a path of *unmarked* edges by introducing new nodes,
3. *unmark* all *marked* edges and stop the construction.

Notice that, where operation 2′ here below was also allowed, then, for any series-parallel construction ϕ of G, there would always exist one (and only one) way of displacing the marks on the edges at the various stages of ϕ in compliance with all preconditions and postconditions dictated by 1, 2, 3, and 2′.

2′. subdivide an *unmarked* edge into a path of *unmarked* edges by introducing new nodes.

Thus, in a sense, a series-parallel construction is canonical when it does not involve applications of rule 2′. Clearly, given any series-parallel construction ϕ of G, by simply anticipating those subdivisions which apply on edges which are not marked, we can derive from ϕ a canonical series-parallel construction of G. This is more formally explained in the following lemma.

Lemma 20. *Let ϕ be a series-parallel construction of a graph G. Then G admits a canonical series-parallel construction ϕ'. Moreover, ϕ' can be derived from ϕ in linear time.*

Proof. By induction on the length $|\phi|$ of ϕ, i.e. on the number of operations involved in ϕ. Assume ϕ is not canonical. This means that at a certain point, ϕ subdivides an unmarked edge ab. Two cases can occur: either ab is an edge of the starting cycle C, in which case we can modify ϕ and consider the construction ϕ' starting from the cycle C' obtained from C by performing in advance the subdivision of the edge ab as later prescribed in ϕ; or ab is an edge introduced by a previous subdivision, in which case we can modify ϕ and consider the construction ϕ' in which the two subdivisions are chained and performed together, without ever introducing the edge ab, at the time the first subdivision was called in ϕ. In both cases, $|\phi'| = |\phi| - 1$, and we can apply induction.

Notice that this proof implies a linear time algorithm to go from ϕ to ϕ'. □

Lemma 21. *Let ϕ be a canonical series-parallel construction of a biconnected bipartite graph G. If ϕ is bipartite, then ϕ is even.*

Proof. Assume on the contrary that ϕ is not even. This implies that at a certain point of the construction ϕ, we have a non-bipartite graph G' containing an odd cycle C'. Being odd, C' contains an edge ab such that a and b have the same color class in G. Notice that the edge ab cannot have been introduced (nor can be later affected) by an operation of type 1, since ϕ is bipartite. Hence, ab is necessarily unmarked. Since a and b are not adjacent in the bipartite graph G, then ϕ will involve a subdivision of the unmarked edge ab. This means that ϕ is not canonical—a contradiction. □

To summarize, we can assume that G is biconnected by Lemma 15 and series-parallel by Theorem 16. In linear time, we can hence find a series-parallel construction ϕ of G. By Lemma 20, we can assume that ϕ is canonical. Now, if ϕ is not bipartite, then we resort to Lemma 19 and return a homogeneous 3-path configuration contained in G. Otherwise, if ϕ is bipartite, then, by Lemma 21, ϕ is even. In this case, we can produce a strongly acyclic orientation of G on the basis of ϕ as previously illustrated.

5.1. A result of Conforti, Cornuéjols and Vušković

Consider an undirected bipartite graph $G = (U, V; E)$ with a weighting $w : E \mapsto \{-1, +1\}$. Following [1], a cycle of G is called *balanced* if the sum of the weights of its edges is a multiple of four. When all cycles are unbalanced, (G, w) is called *totally unbalanced* and w is called a *total unbalancing* of G. A graph G is *totally unbalanceable* when it admits a total unbalancing. In [1], Conforti et al. provided an efficient algorithm to decide whether (G, w) contains a balanced cycle. Moreover, Conforti et al. showed how to decide efficiently whether G is totally unbalanceable or not, and, if yes, to find a total unbalancing. They also gave a forbidden subgraph characterization for totally unbalanceable bipartite graphs.

In this subsection, we observe how these results are essentially equivalent to our results on strongly acyclic digraphs.

Given a bipartite digraph $D = (U, V; A)$, consider the pair (G_D, w_D), where G_D is the underlying graph of D, and to each edge e of G_D we associate $w_e = +1$ if the edge was directed from U to V in D, and $w_e = -1$ otherwise. It should be clear that, where $\overline{C}$ is any round of D and $C = G_{\overline{C}}$ is the corresponding cycle in G_D, then $\sum_{e\in C} w_e \equiv_2 0$, since D is bipartite. Moreover, by virtue of the above definition of the weighting w, $\sum_{e\in C} w_e \equiv_4 2$ if and only if $\overline{C}$ is odd. This is the key to the connection as now developed in the following.

Fact 22. *A bipartite digraph D is strongly acyclic if and only if (G_D, w_D) is totally unbalanced.*

Proof. By definition, (G_D, w_D) is totally unbalanced if and only if G_D contains no cycle whose weight is a multiple of four. Where C is any cycle in G_D, then $\sum_{e\in C} w_e$ is even since G_D is bipartite. Thus, either $\sum_{e\in C} w_e \equiv_4 0$ or $\sum_{e\in C} w_e \equiv_4 2$ occurs. Let $\overline{C}$ be the round of D such that $C = G_{\overline{C}}$. As we further noticed, $\sum_{e\in G_C} w_e \equiv_4 2$ if and only if $\overline{C}$ is odd. By Corollary 13, $\overline{C}$ is odd if and only if $\overline{C}$ is strongly acyclic. In conclusion, (G_D, w_D) is totally unbalanced if and only if $\overline{C}$ is strongly acyclic for every round $\overline{C}$ of D. By Fact 10, this happens if and only if D is strongly acyclic. □

From Fact 22, if follows that our problems about strongly acyclic digraphs are just a restatement of the problems previously treated in [1]. Therefore, our simpler linear time algorithm can be applied also to solve the problems considered in [1]. All the proofs and constructions given in this section are also amenable of a direct translation in that setting.

Acknowledgments

The referees were most helpful in providing useful indications and suggestions. The paper benefited greatly from these.

References

[1] M. Conforti, G. Cornuéjols, K. Vušković, Balanced cycles and holes in bipartite graphs, Discrete Math. 199 (1–3) (1999) 27–33.
[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, McGraw-Hill, Cambridge, MA, Boston, MA, 2001 ISBN: 0-262-03293-7.
[3] G.A. Dirac, A property of 4-chromatic graphs and some remarks on critical graphs, J. London Math. Soc. 27 (1952) 85–92.
[4] J.E. Hopcroft, R.E. Tarjan, Dividing a graph into triconnected components, SIAM J. Comput. 2 (3) (1973) 135–158.
[5] J. Huang, G. MacGillivray, K.L.B. Wood, Pushing the cycles out of multipartite tournaments, Discrete Math. 231 (1–3) (2001) 279–287.
[6] J. Huang, G. MacGillivray, A. Yeo, Pushing vertices in digraphs without long induced cycles, Discrete Appl. Math. 121 (2002) 181–192.
[7] W.F. Klostermeyer, Pushing vertices and orienting edges, Ars Combin. 51 (1999) 65–76.
[8] W.F. Klostermeyer, L. Soltés, Hamiltonicity and reversing arcs in digraphs, J. Graph Theory 28 (1) (1998) 13–30.
[9] G. MacGillivray, K.L.B. Wood, Re-orienting tournaments by pushing vertices, Ars Combin. 57 (2000) 33–47.
[10] D.B. West, Introduction to Graph Theory, Prentice-Hall, Upper Saddle River, NJ, 1996 ISBN: 0-13-227828-6.